

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM SKLADU ELEKTRONICKÉHO OBCHODU

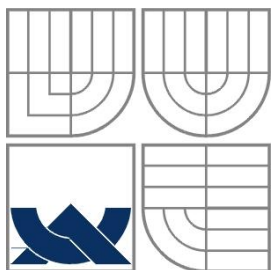
BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

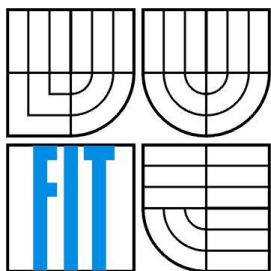
AUTOR PRÁCE
AUTHOR

DANIEL NAGY

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM SKLADU ELEKTRONICKÉHO OBCHODU

INFORMATION SYSTEM FOR ELECTRONIC SHOP WAREHOUSE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DANIEL NAGY

VEDOUČÍ PRÁCE

SUPERVISOR

ING. OTA JIRÁK

BRNO 2009

Abstrakt

Tato bakalářská práce se zabývá informačním systémem pro sklad elektronického obchodu. Cílem bylo vytvořit po důkladné analýze detailní návrh, pak postoupit k implementování jednotlivých částí systému, a vytvořit tak funkční informační systém pro sklad s využitím technologie OLAP a zabezpečit jeho základní funkčnost nad produkční databází. Systém byl implementován v programovacím jazyku PHP, HTML, v relační databáze MySQL a pomocí skriptovacího jazyka JavaScript.

Klíčová slova

Informační systém, sklad elektronického obchodu, databáze, uživatelské rozhraní, PHP, HTML, MySQL, JavaScript, model případu užití, ER diagram, OLAP, statistiky

Abstract

The bachelor's thesis deals with creation of information systems for electronic shop warehouse. After carrying out a thorough analysis, the aim was to form a detailed proposal of an information system, which was followed by the implementation of its particular parts, and then finalized by creation of a functional information system for warehouse using OLAP technology and ensure the basic functionality of the production database.. The system was implemented by the following technologies: PHP, HTML, a relational database MySQL, and JavaScript.

Keywords

Information system, electronic shop warehouse, database, user interface, PHP, HTML, MySQL, JavaScript, use case diagram, ER diagram, OLAP, statistics

Citace

Nagy Daniel: Informační systém skladu elektronického obchodu. Brno, 2009, bakalářská práce, FIT VUT v Brně.

Informační systém skladu elektronického obchodu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Oty Jiráka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
Datum

Poděkování

Tímto bych se chtěl poděkovat mému vedoucímu Ing. Otovi Jirákovi za odborné vedení, cenné rady, konzultace, připomínky, které mi přispěli k dokončení bakalářské práce.

© Daniel Nagy, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod	3
2 Informačný systém	4
2.1 Čo sa rozumie pod pojmom informačný systém?.....	4
2.2 Rozdelenie informačných systémov	4
2.2.1 Systémy dátových transakcií (TPS - Transaction Processing Systems)	5
2.2.2 Klasické informačné systémy na riadenie (MIS - Management Information Systems).....	5
2.2.3 Systémy na podporu rozhodovania (DSS - Decision Support Systems).....	6
2.2.4 Expertné systémy (ES).....	6
2.2.5 Systémy na podporu exekutívy (EIS - Executive Information Systems)	6
2.3 Funkcie informačných systémov	7
3 Použité technológie.....	8
3.1 HTML a CSS	8
3.2 Skriptovací jazyk PHP	9
3.3 Skriptovací jazyk JavaScript	9
3.4 Databázový server MySQL.....	10
3.5 Technológia OLAP	10
3.5.1 Fakty a dimenzie	11
3.5.2 Schéma tabuliek dimenzií.....	13
3.5.3 Úložisko viacrozmerných údajov	14
3.5.4 Technológie na kompresiu objemu použitého diskového priestoru	14
3.5.5 Dôležité OLAP operácie	15
4 Analýza	16
4.1 Požiadavky na informačný systém	16
4.1.1 Hlavné funkcie systému	16
4.2 Modely navrhovaného systému	17
4.2.1 Model prípadov užitia (use case diagram).....	17
4.2.2 ER diagram	19
5 Implementácia	22
5.1 Inštalácia	22
5.2 Vytvorenie databázy	22
5.3 Funkcie informačného systému.....	23
5.3.1 Možnosti zákazníka	23
5.3.2 Možnosti zamestnancov	26

5.4	Funkčnosť OLAP technológií.....	29
5.4.1	Prepojenie tabuliek OLAPu s produkčnou databázou	29
5.4.2	Grafická reprezentácia OLAPu.....	30
6	Záver.....	33
	Literatúra	34
	Zoznam príloh.....	35

1 Úvod

V dnešnej dobe, kedy internet hrá neodmysliteľnú úlohu v živote človeka a firiem, sú informačné systémy veľmi dôležitou súčasťou úspešných firiem a v ich presadzovaní sa na trhu. Bežný človek sa s informačným systémom väčšinou stretáva na strane používateľa, alebo zákazníka, preto by mal byť prístupný a jednoduchý na ovládanie, aby prácu s ním zvládol aj začiatočník. Informačné systémy sú však aj základným pilierom pre správne fungovanie spoločností. Jedna z požiadaviek pre dnešné informačné systémy je poskytnutie čo najväčšieho množstva informácií. Nepochybne medzi ne patrí aj poskytnutie rôznych štatistík. To dosiahneme napríklad použitím technológie OLAP a jej previazaním s produkčnou databázou. Preto som si vybral ako bakalársku prácu informačný systém, ktorý by dokázal spoľahlivo využívať technológiu OLAP.

Druhá kapitola sa zaoberá vysvetlením pojmu informačný systém a jeho popisom. Ďalej ich rozdelením podľa hierarchie riadenia a popisujem tu taktiež funkcie informačných systémov.

V tretej kapitole sa venujem jednotlivým technológiám, ktoré som použil pri tvorbe informačného systému. Nesmie tu chýbať popis HTML a CSS, popis skriptovacieho jazyka PHP a JavaScript. Ako databázový systém je použité MySQL, o ktorom sa tiež podrobnejšie zmienim. Nakoniec sa tu ešte venujem technológii OLAP, ktorá je detailne popísaná.

V štvrtej kapitole sa nachádza analýza navrhovaného systému. Zaoberám sa tu požiadavkami na informačný systém a aké hlavné funkcie by mal plniť. Ďalej tu popisujem modely navrhovaného systému konkrétne model prípadov užitia (use case diagram) a ER diagram.

Piata kapitola sa zaoberá implementáciou, teda popisom funkcií systému. Popisujem tu inštaláciu systému na nový server a vytvorenie databázovej štruktúry. Ďalej je naznačené používateľské rozhranie, prehľad jednotlivých možností zákazníka a možností zamestnancov, ktoré sú doplnené obrázkami pre lepšie pochopenie. Kapitola končí podrobným popisom technológie OLAP, jej prepojenie s produkčnou databázou, použitie triggerov a s grafickou reprezentáciou OLAPu.

Poslednou kapitolou je záver, ktorý obsahuje zhodnotenie dosiahnutých výsledkov.

2 Informačný systém

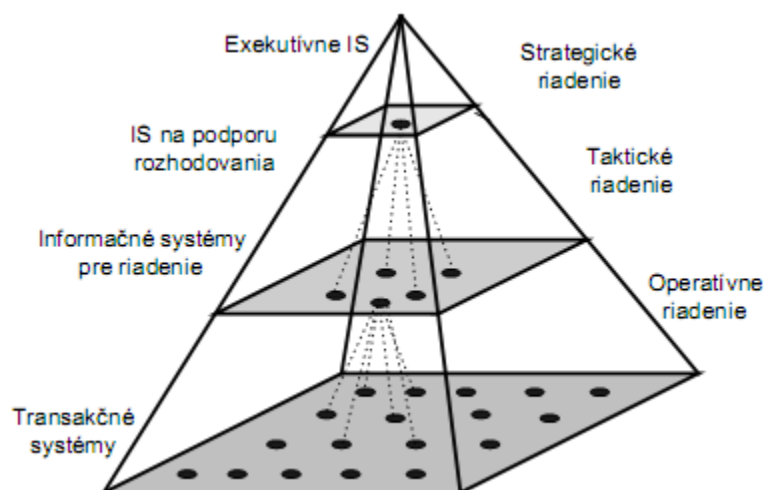
Informačné systémy sú veľmi dôležitou súčasťou dnešného sveta a človek ich deň čo deň využíva, či už v práci, alebo v osobnom živote, aby si uľahčil svoju prácu a život. Firmy, ktoré využívajú informačné systémy sú vo svojej práci efektívnejšie a dokážu rýchlejšie reagovať na vonkajšie podnety.

2.1 Čo sa rozumie pod pojmom informačný systém?

Pod pojmom informačný systém si môžeme predstaviť účelové usporiadanie vzťahov medzi ľuďmi, dátovými zdrojmi a procedúrami ich spracovania, vrátane technologických prostriedkov. Toto usporiadanie zabezpečuje zber, prenos, uchovanie, transformáciu, aktualizáciu a poskytovanie dát na ich informačné využitie ľuďmi. Súčasťou počítačovo podporovaných informačných systémov je aj disponibilné hardvérové a softvérové vybavenie.

2.2 Rozdelenie informačných systémov

Informačné systémy môžeme rozdeliť do viacerých kategórií podľa rôznych hľadísk a radu kritérií. Môžeme ich rozlišovať podľa informačného prostredia, režimu činnosti, kritéria úrovne automatizácie a podľa prevládajúcej funkcie informačného systému. Dôležité kritérium rozdelenia informačného systému je podľa jeho postavenia v systéme riadenia. Podľa tohto kritéria možno informačné systémy rozdeliť do piatich skupín a to sú: Systémy dátových transakcií (TPS - Transaction Processing Systems), Klasické informačné systémy na riadenie (MIS - Management Information Systems), Systémy na podporu rozhodovania (DSS - Decision Support Systems), Expertné systémy (ES), Systémy na podporu exekutívy (EIS - Executive Information Systems) (*Obrázok 2.1*).



Obrázok 2.1: Hierarchické úrovne riadenia

2.2.1 Systémy dátových transakcií (TPS - Transaction Processing Systems)

Tieto systémy nám slúžia na jednoduché spracovanie typických úloh, ako napríklad účtovníctvo, rôzne evidencie, rezervačné či skladové systémy. Dátové operácie sú zväčša vykonané hneď pri ich vložení. S týmto typom informačných systémov sa spravidla stretávame každý deň. Zvláštnym prípadom transakčných systémov pracujúcich v reálnom čase sú informačné systémy na priame riadenie technologických procesov (OLRT – On Line Real Time), ktoré sa vyznačujú minimálnym počtom vstupov a výstupov realizovaných prostredníctvom človeka.

2.2.2 Klasické informačné systémy na riadenie (MIS - Management Information Systems)

Klasické informačné systémy na riadenie sú obvykle založené na rozsiahlej a vhodne organizovanej databáze, ktorá sústreďuje dáta opisujúce základné objekty a procesy vnútri organizácie. Ich hlavným cieľom je podávať rôzne prehľady, napríklad počty objednávok, alebo obrat za určité obdobie. Uľahčujú taktiež prehľad vo výkonnosti jednotlivých pracovníkov a zabezpečujú včasné dodávanie dát pre riešenie rutinných problémov organizácie.

Pre informačné systémy na riadenie nie sú vhodné transakčné systémy, kvôli neustálým hodnotovým zmenám, ktoré sú priebežne aktualizované podľa aktuálnej situácie v podniku a taktiež sa nám zobrazuje len aktuálny stav, kým na podporu rozhodovania obvykle treba určité historické pohľady, napríklad v periódach dní, týždňov, mesiacov, rokov atď. Z údajov s transakčných systémov sa preto robia väčšinou sumarizácie a výbery informácií podľa zvolených kritérií.

2.2.3 Systémy na podporu rozhodovania (DSS - Decision Support Systems)

Pomocou týchto systémov môžeme vykonávať analýzy rovnakých dát, alebo korelácie medzi údajmi v MIS. Okrem toho obsahujú bázu vybraných rozhodovacích modelov, vrátane heuristických. DSS sú vhodným nástrojom na zvýšenie manažérskeho usudzovania a operačných možností. Tam, kde sú DSS použité nad MIS, slúžia na podporu taktického riadenia.

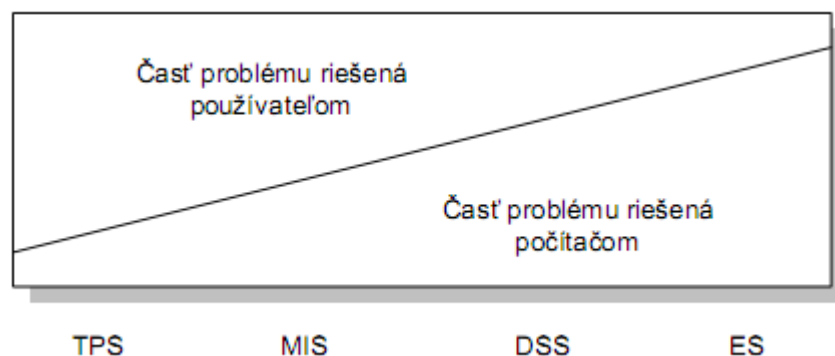
2.2.4 Expertné systémy (ES)

Hlavným cieľom týchto systémov je napodobňovanie postupov, ktoré na základe svojich znalostí a skúseností riešia málo štruktúrované problémy experti. Ich významným komponentom je báza znalostí. Znalosti získané od expertov sú vhodne formalizované a zaznamenané pomocou pravidiel a algoritmov. Expertný systém je schopný užívateľovi riešenie odporúčať a zdôvodniť navrhované riešenie, často s aktívnou účasťou užívateľa na riešení, prípadne na hodnotení.

2.2.5 Systémy na podporu exekutívy (EIS - Executive Information Systems)

Tieto systémy slúžia vrcholovému vedeniu organizácie, ako aj kľúčovým pracovníkom firmy poskytnutím dôležitých informácií, na základe ktorých by mohli uskutočňovať strategické rozhodnutia. Ďalej zabezpečujú väzby a vhodné spracovanie dát z podnikovej databázy (MIS), ale aj z externých databáz a iných dátových zdrojov. Vrcholové vedenie zaujíma viac informácie z okolia podniku, napríklad technické inovácie, konkurencia, politická situácia a pod. Pre súčasný vývoj je charakteristická integrácia EIS a MIS, vrátane podpory rozhodovania. Dôležitá je vysoká vypovedajúca schopnosť výstupu pomocou napríklad grafov.

Podiel ľudského faktora na riešení problémov v informačných systémoch jednotlivých typov zachytáva obrázok 2.2.



Obrázok 2.2: Podiel ľudského faktora na riešení problémov v informačných systémoch

2.3 Funkcie informačných systémov

Cieľom informačných systémov je získavanie, spracovanie a odovzdanie potrebných informácií v potrebnom rozsahu a v požadovanej forme. Ak sa informácie využívajú pri riadení objektov alebo procesov, potom podstatou funkcie informačného systému je na základe definovaných požadovaných výstupných informácií, potrebných pre riadenie, zabezpečiť nevyhnutné vstupné informácie.

Základná funkcia informačných systémov sa realizuje spravidla kombinovaním piatich typických skupín operácií a to sú:

- *Zber dát* – zabezpečenie vstupných informácií.
- *Uchovávanie dát* – a to do vhodne organizovaných dátových štruktúr, ktoré nám umožnia rýchly výber na základe zadaných požiadaviek.
- *Prenos dát* – týka sa prenosu z miesta vzniku na miesto spracovania a podobne z miesta spracovania na miesto ich využitia
- *Prezentácia dát* – záleží na vybraní vhodnej formy spomedzi napríklad textovej, tabuľkovej, grafickej, obrazovej, zvukovej, prípadne kombinácie viacerých foriem.
- *Spracovanie dát* – prebieha na základe presne stanovených postupov, ktorých algoritmus je uložený vo forme programu [1].

3 Použité technológie

Tento informačný systém musí byť dostupný cez celosvetovú sieť Internet prostredníctvom webového prehliadača a to bez ďalšej inštalácie nejakého špecifického softwaru. Ako vhodné riešenie bolo použitie skriptovacieho jazyka vykonávaného na strane serveru, ktorý by bol prepojený s databázou. Po dohode s vedúcim bakalárskej práce som si zvolil skriptovací jazyk PHP a databázový server MySQL, hlavne pre svoju rozšírenosť, nezávislosť na platforme a licenčným podmienkam.

Pri implementácii som využil nasledujúce technológie:

- HTML s použitím kaskádových štýlov CSS
- Skriptovací jazyk PHP
- Skriptovací jazyk JavaScript
- Databázový server MySQL
- Technológia OLAP

3.1 HTML a CSS

Hypertextový značkový jazyk (Hyper Text Markup Language) je značkový jazyk určený na vytváranie webových stránok a iných informácií zobraziteľných vo webovom prehliadači. HTML kladie dôraz skôr na prezentáciu informácií (odseky, fonty, váha písma, tabuľky atď.) ako na sémantiku (význam slov).

Pôvodne bol určený ako veľmi zjednodušená podmnožina jazyka SGML, ktorý sa používa v organizáciách s komplexnými publikačnými požiadavkami, ale neskôr sa stal samostatným štandardom. Špecifikáciu jazyka HTML udržiava W3C (World Wide Web Consortium). Aktuálna verzia je HTML 4.01, ale W3C plánuje nahradiť HTML jazykom XHTML, implementáciou jazyka XML, ktorá zjednoduší prácu prehliadačov aj tvorcov web stránok. Jazyk HTML umožňuje vytvárať dokumenty obsahujúce text, hypertextové odkazy, multimedialny a iný obsah, formuláre, skripty a metainformácie prehliadateľné v tzv. webovom prehliadači.

Jazyk HTML je textový, umožňuje čítanie a upravovanie priamo v textovom editore. Existujú aj špecializované programy na grafické editovanie stránok HTML, napríklad Adobe Dreamweaver alebo Microsoft FrontPage. Ďalšou možnosťou ako vytvárať stránky v jazyku HTML je dynamická (alebo aktívna) tvorba obsahu programom bežiacim na strane servera, napríklad PHP alebo ASP, ktorý na požiadanie vygeneruje stránku a pošle používateľovi. Jazyk HTML sa zvykne používať aj v elektronickej pošte. Na formátovanie HTML stránky sa používa CSS.

Kaskádové štýly alebo CSS (Cascading Style Sheets) je všeobecné rozšírenie (X)HTML. Konzorcium W3C označuje CSS ako jednoduchý mechanizmus na vizuálne formátovanie internetových dokumentov. Štýly umožnili oddeliť štruktúru HTML alebo XHTML od vzhľadu.

Pomocou kaskádových štýlov sa vytvárajú štruktúrované dokumenty, teda oddeľuje sa obsah dokumentu (HTML) od jeho vzhľadu (CSS). Získa sa tým prehľadný a jednoduchý kód. CSS je možné presunúť do externých súborov, zmenší sa tým dátová veľkosť a dá sa jedným súborom zmeniť celý štýl stránky. CSS zaručuje rovnaké vykresľovanie vo všetkých prehliadačoch. Avšak nie vždy je jednoduché dosiahnuť optimalizovaný kód CSS pre celú škálu prehliadačov[2].

3.2 Skriptovací jazyk PHP

PHP (Hypertext Preprocessor) je populárny open source skriptovací programovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií (na strane servera) a pre vývoj dynamických webových stránok. Medzi známe aplikácie založené na PHP patrí napríklad phpBB a MediaWiki, software na ktorom beží Wikipédia.

PHP bolo inšpirované jazykmi podporujúcimi štruktúrované programovanie, najviac vlastností prebralo od jazyka C a jazyka Perl. V neskorších verziách bolo rozšírené o možnosť používať objekty. Jedna zo zaujímavých vlastností PHP je, že umožňuje viac ako bežný skriptovací jazyk. Vďaka modulárnemu návrhu možno PHP používať aj na vývoj aplikácií s užívateľským rozhraním (GUI).

PHP dokáže spolupracovať s relačnými databázami, ako napríklad MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL a SQLite, pričom si stále zachováva jednoduchú a priamočiaru syntax. PHP beží na takmer všetkých najrozšírenejších operačných systémoch, vrátane UNIXu, Linuxu, Windowsu a Mac OS X. Spolupracuje s najrozšírenejšími webovými servermi. Architektúra Linux, Apache, MySQL, PHP (zaužívaná skratka je LAMP) sa stala veľmi obľúbenou v internetovom odvetví [3].

3.3 Skriptovací jazyk JavaScript

JavaScript, je skriptovací programovací jazyk. Jazyk je používaný najmä pri tvorbe webových stránok. Pôvodne ho vyvíjal Brendan Eich zo spoločnosti Netscape Communications pod názvom Mocha, neskôr pod menom LiveScript. Pred uvedením na verejnosť bol premenovaný na „JavaScript“, najmä pre vtedajšiu popularitu jazyka Java. Aj na základe jeho názvu je rozšírený názor, že syntax Javascriptu sa podobá Jave, v skutočnosti bol jeho tvorca najviac inšpirovaný jazykom Self.

Neexistuje žiadny vzťah medzi jazykmi Java a JavaScript, ich najväčšia podobnosť je v syntaxi založenej na spoločnom predkovi - jazyku C. Microsoft vyvinul jazyk s podobnou syntaxou pod názvom JScript, ktorý ale nedodržuje špecifikáciu ECMAScript. Rozdiely v implementácii majú za

následok čiastočnú neprenosnosť kódu medzi internetovými prehliadačmi používajúcimi JScript (Microsoft Internet Explorer) a JavaScript (Mozilla Firefox). Výhodou tohto skriptovacieho jazyka je, že zo strany servera nezaťažuje, všetko sa deje na strane klienta. Pre správne fungovanie treba mať povolený tento skriptovací jazyk v prehliadači.

V súčasnosti je dostupné množstvo implementácií virtuálnych strojov jazyka JavaScript, proprietárnych i s otvoreným zdrojovým kódom, pričom niektoré z týchto implementácií je možné používať ako knižnice do iných programov a mať tak podporu tohto jazyka v prakticky ľubovoľnej aplikácii. V poslednej dobe sa začínajú objavovať implementácie, ktoré prevádzajú JavaScriptové programy priamo do procesorových inštrukcií a teda ich môžu priamo spustiť na danej architektúre bez potreby dodatočného virtuálneho stroja.

3.4 Databázový server MySQL

MySQL je slobodný a otvorený viacvláknový, viac užívateľský SQL relačný databázový server. Je podporovaný na viacerých platformách (ako Linux, Windows či Solaris) a je implementovaný vo viacerých programovacích jazykoch ako PHP, C++ či Perl. Databázový systém je relačný typu DBMS (database management system). Každá databáza je v MySQL tvorená z jednej alebo z viacerých tabuliek, ktoré majú riadky a stĺpce. V riadkoch sa rozoznávajú jednotlivé záznamy, stĺpce udávajú dátový typ jednotlivých záznamov, pracuje sa s nimi ako s poľami. Práca s MySQL databázou je vykonávaná pomocou takzvaných dotazov, ktoré vychádzajú z programovacieho jazyka SQL (Structured Query Language).

Pri práci s DBMS programom sa používa osvedčený model klient/server. V roli serveru vystupuje DBMS program, ktorý je spustený a čaká na požiadavky od klienta. Na tieto požiadavky potom odpovie a reaguje na nich. Pretože DBMS program vystupuje v roli serveru, hovorí sa mu databázový server. V roli klienta pre databázový server môže ako aplikácia vystupovať i PHP. To znamená, že naše PHP programy môžu obsahovať príkazy zapísané v programe SQL a PHP môže spracovávať výsledky SQL po prevedení na databázový server.

MySQL AB je softwarová spoločnosť, vyvíjajúca databázový systém MySQL. Robí ho dostupný ako voľný software pod licenciou GNU General Public License (GPL), no jednako tiež ponúkajú členstvo MySQL Enterprise, vhodné pre pracovníkov a dvojaké licencovanie pod bežným konceptom licencie pre použitie iné než uvedené (alebo nekompatibilné s podmienkami) v GPL [4].

3.5 Technológia OLAP

Technológia OLAP (Online Analytical Processing) slúži k spracováaniu údajov v dátovom sklade do podoby, z ktorej sú manažéri a analytici schopní získavať informácie a podľa nich robiť firemné rozhodnutia ohľadne predaja či marketingu.

OLAP je teda voľne definovaný poriadok princípov, ktoré poskytujú dimenzionálny rámec pre podporu rozhodovania. Autorom názvu OLAP bol Dr. E. F. Codd, ktorý tak definoval nástroj pre správu podnikových dát na osobnom počítači. Okrem definície pojmu OLAP je pomerne známe tiež tzv. dvanásť pôvodných pravidiel OLAP (Od Dr. E. F. Codd [5]):

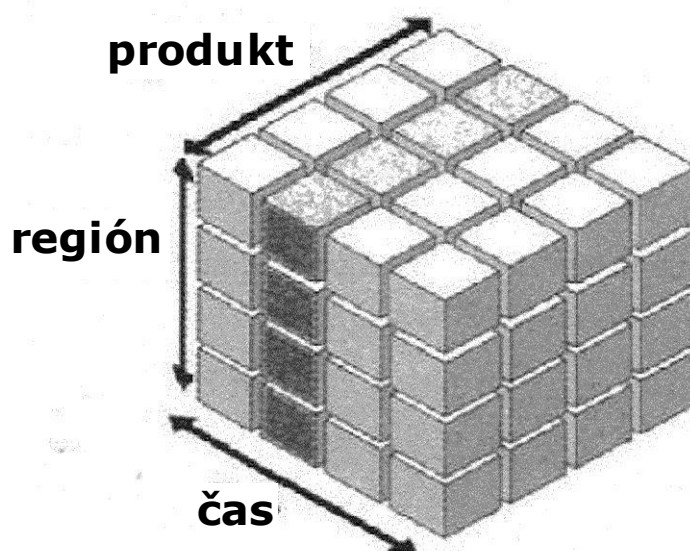
1. *Viacrozmerný konceptuálny pohľad*, ktorý najlepšie zodpovedá podnikateľským potrebám v zmysle využitia pre analýzu zhromaždených údajov.
2. *Transparentnosť*, ktorá je potrebná pre používateľa vzhľadom na možnosti využitia vo vhodných a známych prostrediach.
3. *Dostupnosť*, kedy prístup OLAP systému by mal byť len k údajom, ktorých sa analýza týka.
4. *Konzistentné vykazovanie*, čo znamená, že s nárastom údajov v databáze by nemalo nastať výrazné zníženie výkonu systému.
5. *Architektúra klient-server*, systém musí odpovedať tejto architektúre s prihliadnutím k maximálnej cene a výkonu, flexibilita a interoperabilita.
6. *Generická dimenzionalita*, naznačujúca, že každá dimenzia údajov musí byť rovnaká v štruktúre aj operačných schopnostiach.
7. *Dynamické ošetrovanie riedkych matic*, systém by mal vedieť použiť svoje fyzické schéma na príslušný analytický model, ktorý optimalizuje.
8. *Podpora pre viac užívateľov*, systém spravujúci dáta musí byť schopný podporovať určitú skupinu užívateľov pracujúcich súčasne na konkrétnom modeli.
9. *Neobmedzené krížové dimenzionálne operácie*, systém musí dokázať rozoznať dimenzionálnu hierarchiu a samostatne vykonávať kalkulácie v rámci dimenzií a medzi nimi.
10. *Intuitívna manipulácia s údajmi*, toto pravidlo definuje možnosť prepísania ciest na detailnú úroveň a späť, všetky manipulácie by mali byť v jednoduchom používateľskom prostredí.
11. *Flexibilné vykazovanie*, musí existovať schopnosť usporiadať riadky, stĺpce a bunky pre prezentáciu analytických zostáv.
12. *Neobmedzené dimenzie a úrovne agregácie*, v závislosti na požiadavkách môže mať analytický model viac dimenzií, pričom každá z nich môže mať viacnásobné hierarchie. Systém by ich počet nemal nijak umelo obmedzovať.

3.5.1 Fakty a dimenzie

V OLAP sa zvyčajne používajú tzv. kocky, čo sú viacrozmerné dátové štruktúry, ktoré sú výsledkom agregácie a analýzy údajov. Každá táto kocka je vytvorená na základe dvoch druhov údajov, a to faktami a dimenziami. Týchto dimenzií, na rozdiel od geometrickej kocky, môže byť aj viac ako tri. Samotné údaje sa nachádzajú v prienikoch jednotlivých dimenzií (*obrázok 3.1*).

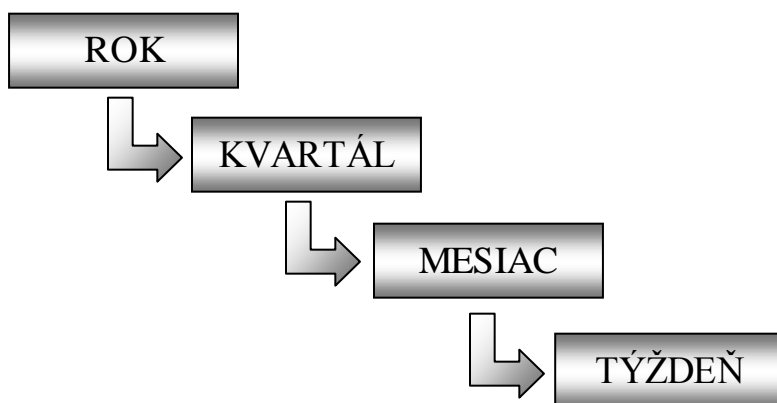
Fakty sú numerické merné jednotky obchodovania. Tabuľka faktov je najväčšia tabuľka v databáze a obsahuje veľký objem dát. Tabuľky faktov a dimenzií môžu vytvárať určité schéma, napríklad hviezdicové schéma, alebo schéma snehovej vločky. Hviezdicové schéma zväčša obsahuje len jednu tabuľku faktov, iné ich však môžu obsahovať viacero. Pri vytváraní tabuľky faktov sa môže využiť iných faktov a vytvoriť tak merné jednotky. Tie sa môžu uložiť v tabuľke faktov prípadne vyvolať na účely vykazovania.

Dimenzie obsahujú logicky, alebo hierarchicky usporiadané údaje. Tabuľky dimenzií sú väčšinou menšie ako tabuľky faktov a dáta v nich sa tak často neaktualizujú. Tabuľky dimenzií vysvetľujú príčinu a spôsob obchodovania a transakciu prvkov. Dimenzie obsahujú relatívne stabilné dáta. Najčastejším typom dimenzií sú časové, produktové a geografické (obrázok 3.2).



Obrázok 3.1: Kocka OLAP s časovou, produktovou a geografickou dimenziou

Typickou štruktúrou pre tabuľky dimenzií je stromová, môže byť vytvorená na základe informácií z časových, produktových, alebo geografických údajov.

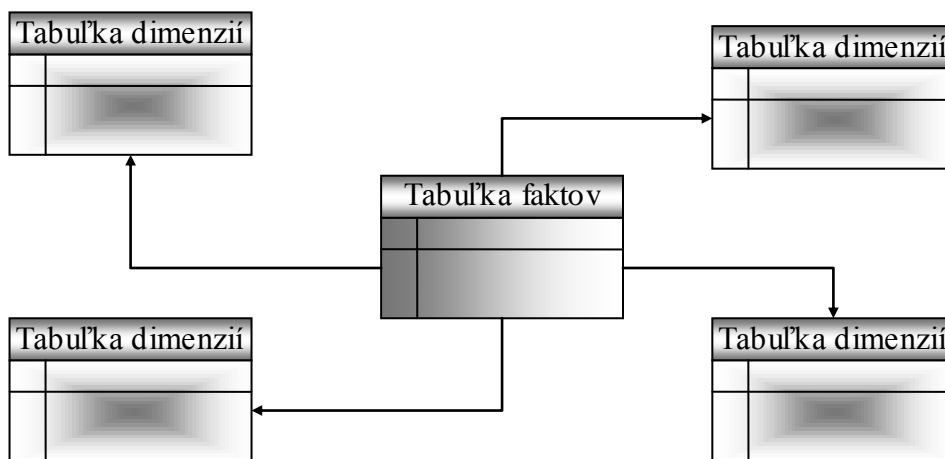


Obrázok 3.2: Hierarchia časovej dimenzie

3.5.2 Schéma tabuliek dimenzií

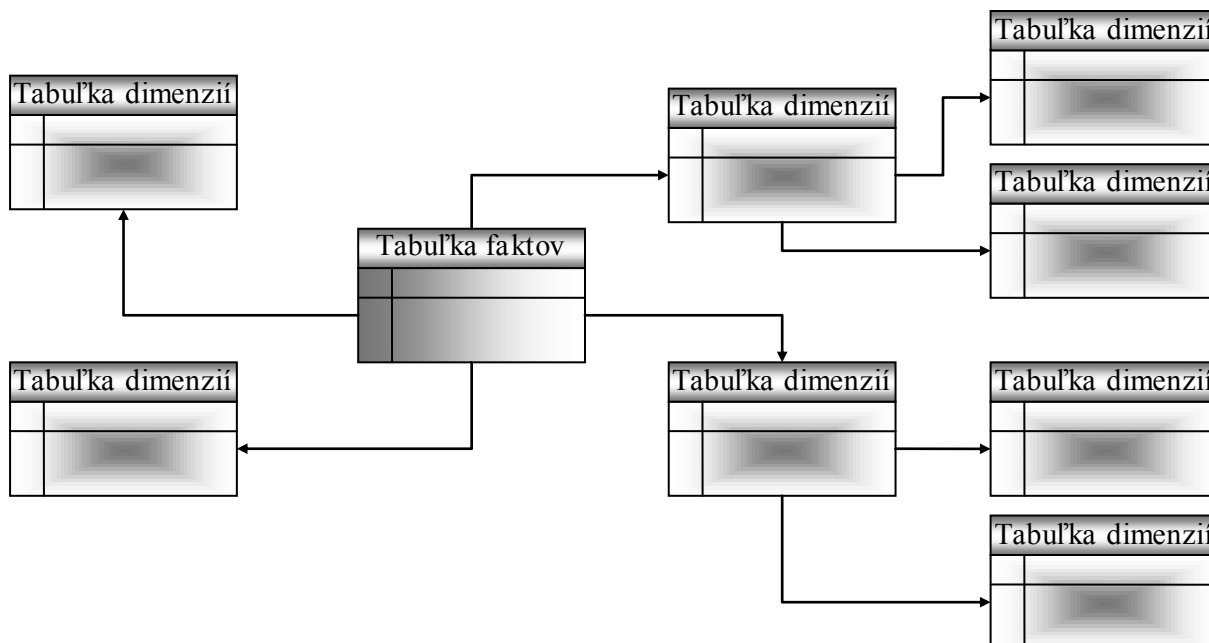
Kocka je tvorená na základe určitého dimenzionálneho modelu, ktorý má určité topologické usporiadanie. Najčastejším usporiadaním býva hviezdicové schéma, alebo schéma snehovej vločky.

Hviezdicové schéma obsahuje tabuľku faktov s cudzími kľúčmi, ktoré sa vzťahujú k primárnym kľúčom v dimenziách (obrázok 3.3). Hviezdicové schéma nemá normalizované dimenzie ani relačné prepojenia medzi tabuľkami dimenzií. Medzi pozitívne vlastnosti tohto usporiadania patrí jeho vysoký opytovací výkon. Ten je zrejmý z toho, že tabuľky nie sú štandardizované, preto sa relačné tabuľky nemusia spájať.



Obrázok 3.3: Hviezdicové schéma (star schema)

Schéma snehovej vločky obsahuje niektoré dimenzie zložené z viacerých relačne zviazaných tabuliek (obrázok 3.4). Tento model umožňuje rýchlejšie zavedenie údajov do normalizovaných tabuliek, ale má podstatne nižší opytovací výkon, lebo obsahuje väčšie množstvo spojení tabuliek.



Obrázok 3.4: Schéma snehovej vločky (snowflake schema)

3.5.3 Úložisko viacrozmerných údajov

Dátovým modelom viacrozmernej databázy je viacrozmerná kocka. Každá kocka má niekoľko dimenzií, ktoré určujú rozmernosť kocky. Platí pravidlo, že čím má viacrozmerná kocka viac rozmerov, tým je náročnejšia na úložnú kapacitu. Jednotlivé záznamy sa v kocke nachádzajú na priesečníkoch dimenzií. Môže však nastať aj situácia, kedy na priesečníku dimenzií nie je žiadny záznam, takéto kocky sú nazývané riedke.

3.5.4 Technológie na kompresiu objemu použitého diskového priestoru

V praxi sa pri viacrozmerných databázach používajú rôzne technológie na kompresiu objemu použitého diskového priestoru [5].

3.5.4.1 Viacrozmerný OLAP (MOLAP)

Táto technológia získava dáta buď z dátového skladu alebo z operačných zdrojov. Mechanizmus OLAP ukladá tieto analytické dáta do vlastných dátových štruktúr a sumárov. Počas tohto procesu sú prepočítavané výsledky a to v množstve, ktoré je z časového hľadiska možné. Údaje sa ukladajú do úložiska MOLAP ako vopred vypočítané polia, kde v jednotlivých poliach sú potom uložené hodnoty dát. Kvôli menšiemu zaťaženiu siete sa občas prevádzajú niektoré údaje priamo ku klientovi. Nevýhodou tejto technológie je redundancia údajov, čo má za následok rastúce nároky na kapacitu úložiska.

3.5.4.2 Relačný databázový OLAP (ROLAP)

V relačnom online analytickom spracovaní údajov sú údaje získavané z relačného dátového skladu. Tieto údaje sa užívateľovi predkladajú ako viacrozmerný pohľad, kedy sa dáta a metadáta ukladajú ako záznamy v relačnej databáze. Server ROLAP využíva metadáta na generovanie príkazov v jazyku SQL (Structured Query Language), potrebných pre získanie dát požadovaných používateľom. Keďže dáta zostávajú uložené v relačných databázach, nenastáva problém s redundanciou dát.

3.5.4.3 Hybridný OLAP (HOLAP)

Táto technológia je kombináciou MOLAP a ROLAP, keď údaje zostávajú v relačných tabuľkách a spočítané agregácie sú ukladané do viacrozmerných štruktúr. Pri vyžiadaní sa využíva na vyberanie údajov z viacrozmernej pamäte cache. U hybridnej technológie si relačná databáza ukladá veľa detailných údajov a viacrozmerný model ukladá sumárne dáta.

3.5.5 Dôležité OLAP operácie

Dôvodom vzniku dátového skladu je možnosť jednoducho uskutočniť analýzy zhromaždených údajov. Nasledujú základné operácie umožňujúce tieto analýzy:

- *Drill-down*: umožňuje užívateľovi na zvolenom atribúte danej agregáčnej úrovne nastaviť nižšiu agregáčnú úroveň.
- *Roll-up*: táto operácia je opak drill-down, kedy sa pre vybraný atribút danej agregáčnej úrovne nastavuje vyššia agregáčná úroveň.
- *Pivoting*: umožňuje meniť pohľad na dáta na úrovni prezentácie obsahu dátového skladu
- *Slice & Dice*: jedná sa o výber projekcie, čo je v podstate zatajovanie a odkrývanie jednotlivých atribútov (redukovanie rozmerosti dát) [5].

4 Analýza

Táto kapitola sa týka analýzy a návrhu informačného systému skladu. Sú tu zhrnuté požiadavky kladené na výsledný systém, jeho najdôležitejšie funkcie a taktiež popísané modely informačného systému. Posledná časť kapitoly sa týka čistého návrhu a ich modelov. Týka sa to modelu prípadov užívania (use case diagram) a ER diagramu, ktoré sú podrobne opísané aj na daný navrhovaný systém.

4.1 Požiadavky na informačný systém

Pred tvorbou informačného systému je potrebné si zistiť, aké sú hlavné požiadavky na výsledný systém a aké funkcie by mal plniť. Samozrejme som sa touto otázkou zaoberal a vyskúšal som viacero systémov pri čom som sa mohol inšpirovať. V informačnom systéme by sa mal ľahko orientovať aj človek, ktorý by v ňom pracoval prvýkrát, preto by mal byť jednoduchý, nenáročný na schopnosti užívateľa a hlavne prehľadný.

4.1.1 Hlavné funkcie systému

Je samozrejmosťou, že informačné systémy využívajú používatelia s rôznymi právami prístupu k informáciám. Jednotlivé funkčnosti systému sú prístupné rôzni používateľom, na základe toho, ku ktorej roli je systémom používateľ priradený. Môj systém dokáže pracovať so štyrmi rôznymi typmi účtov, ktorí môžu vykonávať rôzne operácie:

- ***Zákazník:***
 - pred vstupom do systému je povinný sa zaregistrovať
 - prezerať tovar, prípadne si ho objednať
 - upraviť osobné údaje, ktoré boli povinné zadať pri registrácii
 - editovať objednávku
 - stornovať objednávku
- ***Správca objednávok:***
 - objednať nový tovar na sklad
 - editovať objednávku
 - stornovať objednávku
 - skontrolovať stav objednávky
- ***Skladník:***
 - kontrolovať príjem objednávok
 - kontrolovať výdaj objednávok

- triedenie objednávok
- **Vedúci:**
 - má rovnaké úkony ako správca a skladník
 - zamestnať zamestnanca
 - prepustiť zamestnanca

4.2 Modely navrhovaného systému

Pred implementáciou je potrebné navrhnuť si dva modely a to sú model prípadov užitia (use case diagram) a ER diagram, pomocou ktorých vieme pochopiť podstatu problému a aj spôsob jeho riešenia.

4.2.1 Model prípadov užitia (use case diagram)

Model prípadov (*obrázok 4.1*) užitia patrí medzi modely popisujúce funkcionálnu architektúru systému a to ako bude vlastne systém fungovať z pohľadu používateľa. Práve preto je tento typ modelu vhodný pre komunikáciu s používateľom. Služi pre základné vymedzenie hraníc medzi systémom a jeho okolím. Komponentmi tohto modelu sú:

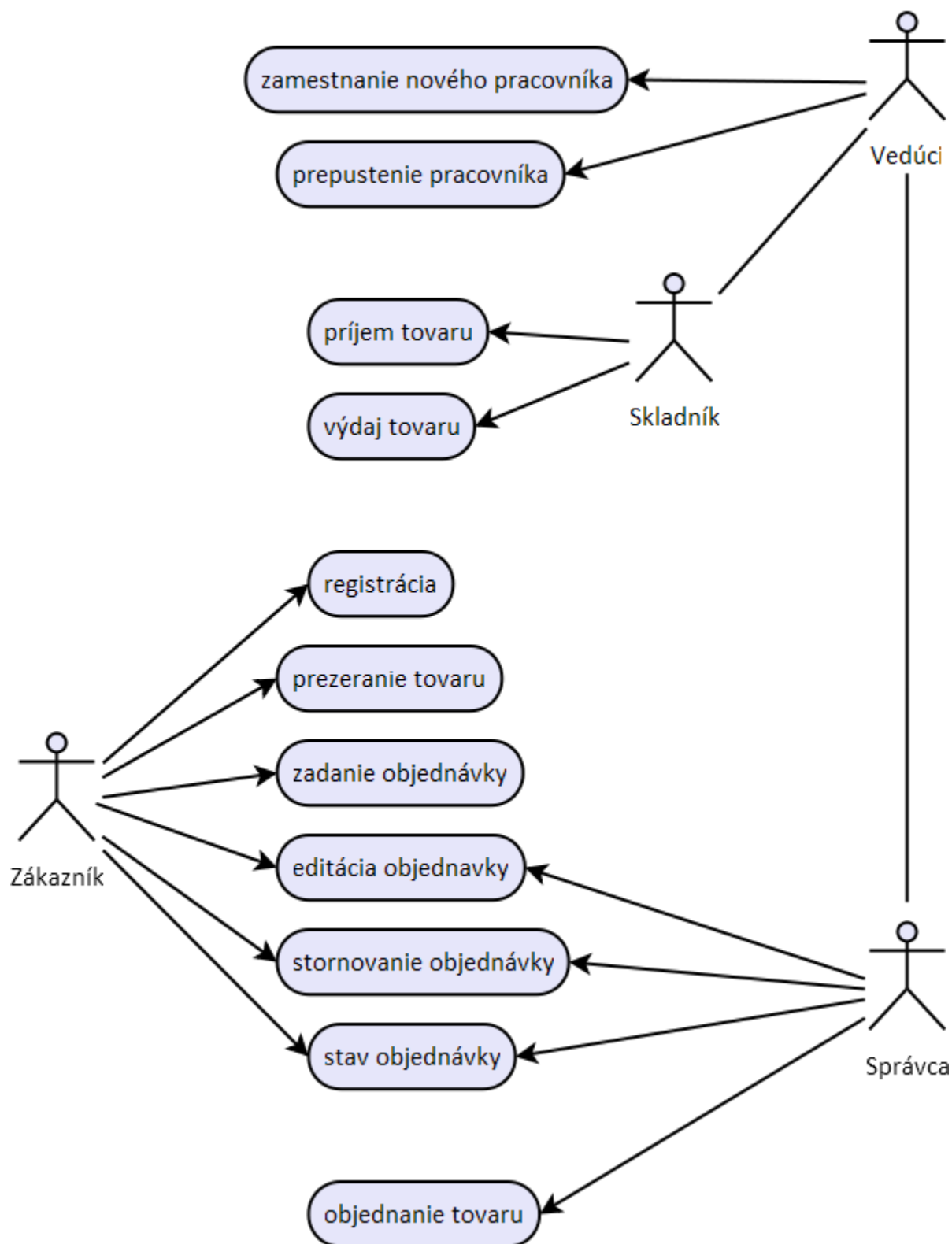
- Zákazník
- Správca objednávok
- Skladník
- Vedúci

K systému teda pristupujú štyri typy používateľov. Prvý je zákazník, ktorý je povinný sa zaregistrovať, keď sa chce prihlásiť do systému. Keď sa zákazník zaregistruje stáva sa z neho registrovaný používateľ. Ten môže prezerat' tovar, ktorý sklad ponúka a môže si ho objednať. Môže taktiež svoje objednávky editovať, alebo stornovať. Zákazník si taktiež môže zmeniť osobné údaje, ktoré zadával pri registrácii. Môže si zmeniť heslo a adresu na ktorú mu bude prípadný tovar z objednávky dodaný.

Druhou osobou, ktorá môže do systému pristupovať je správca objednávok, ten môže prezerat', editovať, alebo stornovať objednávky zákazníkov a prípadne objednať tovar, ktorý už nie je na sklade.

Tretou osobou, ktorá sa dostane do systému je skladník. Jeho práva sú obmedzené, len na kontrolu príjmu tovaru do skladu a výdaju objednávok zo skladu. Taktiež ešte kontroluje triedenie objednávok podľa doručovacieho obvodu.

Poslednou osobou, ktorá môže do systému pristupovať je vedúci skladu, ktorý môže prevádzkať rovnaké úkony ako správca objednávok a skladník a navyše ešte môže prijať nového zamestnanca, alebo prepustiť stávajúceho zamestnanca.



Obrázok 4.1: Model prípadov užitia (use case diagram)

4.2.2 ER diagram

Klasickým spôsobom prezentácie modelovaného sveta je ER diagram. Je to grafická reprezentácia entít, vzťahov a obmedzení, ktoré tvoria daný návrh. Jeho základom je prevedenie komplexných štruktúr modelovanej skutočnosti do dvojrozmerných tabuliek a nájdenie vzťahov medzi nimi. Efektívny dátový model presne a úplne popisuje a vyhovuje nárokom zadania a je použiteľný pre tvorcu databázy, eliminuje redundanciu dát a je nezávislý od hardvéru a softvéru. Práve ľahká transformovateľnosť do tabuliek ho robí vhodným pre relačné databázy. Pre návrhára to je veľmi užitočné hlavne pre validáciu správnosti návrhu. Neexistujú obecne platné konvencie pre nakreslenie ER diagramov a preto môžu existovať rozdiely a rôzne typy ER diagramov. ER diagram obsahuje entity, atribúty a vzťahy:

- **Entita (entity)** – objekt reálneho sveta
- **Atribút (Attribute)** – vlastnosť entity, ktorá nás zaujíma
- **Vzťah (relationship)** – asociácia medzi niekoľkými entitami

ER diagram reprezentuje logické vzťahy medzi entitami alebo objektmi. Má nezastupiteľnú úlohu nielen pri návrhu databázových aplikácií, ale aj pri ich optimalizácii a odstraňovaní chýb. Pri vytvorení ER diagramu je treba rozhodnúť, ktoré entity, vzťahy a obmedzenia použijeme pre modelovanie fyzického systému. U jednoduchého príkladu môže toto rozhodnutie byť jednoduché, ale pre praktické informačné systémy ide čiastočne i o umenie a skúsenosti.

Špeciálne druhy vzťahov predstavujú agregácie, kompozície a generalizácie:

- **Agregácie (aggregation)** - je druh vzťahu, keď jedna trieda je súčasťou inej triedy.
- **Kompozícia (composition)** - je silnejší druh agregácie. Pri kompozícii je časť priamo závislá na svojom celku, zaniká zmazaním celku a nemôže byť súčasťou viac ako jedného celku
- **Generalizácia (generalization)** - druh vzťahu, keď jedna trieda je všeobecným zavedením vlastností inej triedy (iných tried)

4.2.2.1 ER diagram informačného systému

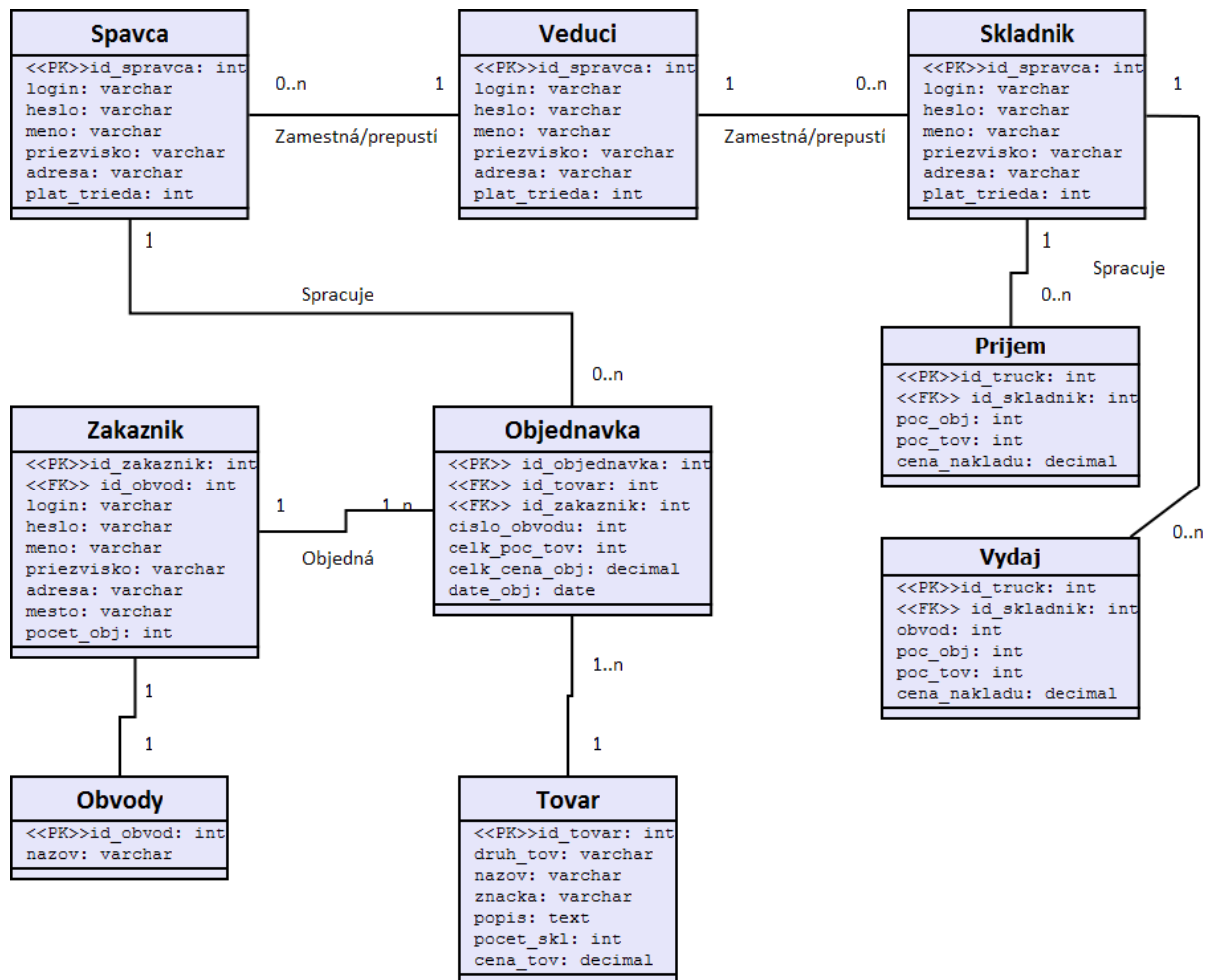
ER diagram môjho informačného systému je zobrazený na *obrázku 4.2* a obsahuje nasledujúce entity, tabuľky dimenzií OLAP a tabuľku faktov OLAP:

- **Objednávka** – Tabuľka obsahuje všetky zadané objednávky. Primárny kľúč *id_objednavka* jednoznačne určuje zadanú objednávku. Ďalej obsahuje cudzie kľúče *id_tovar*, *id_zakaznik* a *id_spravca*, ktoré k objednávke zaraďujú konkrétny tovar, zákazníka a správcu, ktorý objednávku spracuje. Ďalšími atribútmi sú *cislo_obvodu*, ktoré reprezentuje kraj z ktorého pochádza zákazník, *celk_poc_tov*, ktorý predstavuje počet objednaných výrobkov, *celk_cena_obj* predstavuje cenu objednávky a *date_obj* predstavuje dátum zadania objednávky. Entita objednávka je vo vzťahu so zákazníkom

s kardinalitou 1:0..n, čo znamená, že zákazník môže mať zadaných viacero objednávok, ale nemusí mať zadanú žiadnu. Tiež je vo vzťahu s ďalšími entitami, ako správca, alebo tovar.

- **Tovar** – Táto tabuľka obsahuje informácie o ponúkanom tovare. Primárny kľúč *id_tovar* presne identifikuje tovar. Atribúty *nazov*, *znacka*, *popis* obsahujú bližšie informácie o tovare, *pocet_skl* predstavuje počet uskladnených kusov daného tovaru, *cena_tov* obsahuje cenu tovaru.
- **Zákazník** – Táto tabuľka obsahuje informácie o zákazníkoch. Primárnym kľúčom je *id_zakaznik* jednoznačne určuje zákazníka. Obsahuje cudzí kľúč *id_obvod*, ktorý k zákazníkovi zaraďuje kraj, z ktorého pochádza. Ďalšími atribútmi sú *login*, *heslo*, *meno*, *priezvisko*, *adresa*, *mesto*, ktoré obsahujú osobné údaje zákazníka a *pocet_obj* predstavuje počet zadaných objednávok. Entita zákazník je vo vzťahu s entitou obvod s kardinalitou 1:1 a tiež s entitou objednávka s kardinalitou 1:0..n.
- **Obvody** – Tabuľka obsahuje primárny kľúč *id_obvod*, ktorý určuje kraj a atribút *nazov*, ktorý obsahuje názov kraja.
- **Správca** - Tabuľka obsahuje informácie o zamestnaných správcach skladu. Primárnym kľúčom je *id_spravca* jednoznačne určuje správcu. Ďalšími atribútmi sú *login*, *heslo*, *meno*, *priezvisko*, *adresa*, ktoré obsahujú osobné údaje správcu a *plat_trieda* predstavuje zaradenie zamestnanca do platovej triedy.
- **Skladník** – Táto tabuľka obsahuje informácie o zamestnaných skladníkoch. Primárny kľúč *id_skladnik* jednoznačne identifikuje skladníka. Ďalšími atribútmi sú *login*, *heslo*, *meno*, *priezvisko*, *adresa*, ktoré obsahujú osobné údaje skladníka a *plat_trieda* predstavuje zaradenie zamestnanca do platovej triedy. Entita skladník je vo vzťahu s entitou príjem a s entitou výdaj s kardinalitou 1:0..n.
- **Vedúci** - Tabuľka obsahuje informácie o zamestnaných vedúcich skladu. Primárnym kľúčom je *id_veduci* jednoznačne určuje vedúceho. Ďalšími atribútmi sú *login*, *heslo*, *meno*, *priezvisko*, *adresa*, ktoré obsahujú osobné údaje vedúceho a *plat_trieda* predstavuje zaradenie zamestnanca do platovej triedy.
- **Príjem** – Obsahuje informácie o prichádzajúcom tovare do skladu. Primárny kľúč *id_truck* reprezentuje číslo prostriedku, ktorým bol tovar dodaný, cudzí kľúč *id_skladnik* identifikuje skladníka, ktorý tovar prevzal. Ďalšími atribútmi sú: *poc_obj*, *poc_tov* a *cena_nakladu* reprezentujú počet naložených objednávok, počet naloženého tovaru a celkovú cenu nákladu.
- **Výdaj** – Obsahuje informácie o vydávanom tovare zo skladu. Primárny kľúč *id_truck* reprezentuje číslo prostriedku, ktorým je tovar odvážaný, cudzí kľúč *id_skladnik* identifikuje skladníka, ktorý tovar vydal. Ďalšími atribútmi sú: *poc_obj*, *poc_tov*

a *cena_nakladu* reprezentujú počet naložených objednávok, počet naloženého tovaru a celkovú cenu nákladu.



Obrázok 4.2: ER diagram informačného systému.

5 Implementácia

V nasledujúcej kapitole sa budem zaoberať detailným popisom implementácie jednotlivých častí systému. Opísal som tu riešenia častí systému a aké som pri implementácii použil techniky a algoritmy. Začnem popisom implementácie databázovej štruktúry systému, ďalej popisom aplikačnej a klientskej vrstvy. Bližšie tu ukážem aj operácie s dátami uloženými v databáze. Vysvetlím jednotlivé funkcie systému pre používateľov, aj pre administrátorov. Nakoniec sa pokúsím pomocou obrázkov naznačiť riešenie užívateľského rozhrania.

5.1 Inštalácia

Na inštaláciu je potrebný webový server s podporou technológie PHP, ktorý musí mať povolený modul MySQL pre prácu s databázou a musí mať prístup k databáze MySQL. Na vytvorenie pripojenia k databáze MySQL je potrebné nastaviť prístupové údaje a to: meno servera, prístupové meno do databázy, prístupové heslo do databázy a názov databázy v súbore *connection.php*.

```
$db = mysql_connect('meno servera', 'prístupové meno do databázy', 'prístupové  
heslo do databázy');  
if (!$db) die('nemožem sa pripojiť '.mysql_error());  
if (!mysql_select_db('názov databázy', $db)) die('databáza nie je dostupná  
' .mysql_error());
```

5.2 Vytvorenie databázy

Na vytvorenie databázovej štruktúry posluží skript, ktorý sa nachádza medzi zdrojovými súborami s názvom *databaza.txt*. Tento skript naplní databázu potrebnými tabuľkami a aj údajmi na správu systému, ktorý je potom funkčný a pripravený na použitie. Samotná databáza sa skladá z jedenástich tabuliek, ktoré sú implementované podľa ER diagramu (viď. kapitola 4.2.2.1 ER diagram informačného systému).

Každá entita reprezentuje samostatnú tabuľku a vzťahy medzi nimi sú riešené pomocou cudzích kľúčov. Kľúč sa pri návrhu zložitejších databáz. Existujú dva typy kľúčov: primárny a cudzí. Každá tabuľka v mojej implementácii obsahuje jeden primárny kľúč a ten je obvykle spojený ako cudzí kľúč s inou tabuľkou. Primárny kľúč tabuľky odkazuje na záznam a musí spĺňať tri pravidlá:

- Hodnota musí byť vždy zadaná.
- Hodnota musí byť v celej tabuľke jedinečná.
- Hodnota sa nikdy nesmie zmeniť.

Primárne kľúče v mojom systéme majú zadaný modifikátor *AUTO_INCREMENT*, pomocou ktorého sa automaticky generujú unikátne hodnoty primárneho kľúča. Tabuľky v mojej databáze obsahujú nasledujúce dátové typy:

- *INT* – reprezentuje celé číslo
- *DECIMAL* – je to číslo typu *double*, ktoré je uložené ako reťazec s pevnou desatinnou čiarkou
- *VARCHAR* – slúži k ukladaniu reťazcov
- *TEXT* – môžeme doň uložiť reťazec o dĺžke 65535 znakov.
- *DATE* – je to dátový typ používaný pri dátume.

5.3 Funkcie informačného systému

V tejto časti kapitoly popíšem jednotlivé implementované funkcie môjho systému, na úrovni aplikačnej vrstvy. Vysvetlím funkcie, ktoré môžu vykonávať zákazníci a ktoré zamestnanci (*vid'. kapitola 4.2.1 Model prípadov užívania*). Funkcie sú implementované v jazyku PHP (*opis tohto jazyka vid'. kapitola 3.2 Skriptovací jazyk PHP*) a doplnené sú skriptovacím jazykom JavaScript pre vylepšenie používateľského rozhrania.

5.3.1 Možnosti zákazníka

Zákazník, ktorý si chce objednať tovar z veľkoskladu sa musí registrovať. Pri registrácii sú od neho požadované osobné údaje, ktoré je povinný vyplniť (*obrázok 5.1*). Po odoslaní registračného formulára sa kontroluje správnosť údajov. Skontroluje sa, či sú všetky políčka vyplnené a následne sa skontroluje správnosť políček. Keď systém narazí na chybné vyplnené políčko, tak na to upozorní chybovým hlásením. Jednou z dôležitých kontrol je *login* zákazníka, kedy je potrebné skontrolovať jeho jedinečnosť. Každý používateľ musí mať jedinečný *login* pre prihlásenie do systému. Ak by došlo k zhode *loginu*, tak na to systém upozorní chybovým hlásením. Ešte je potrebné spomenúť, že *heslo* je z bezpečnostných dôvodov do databázy uložené pomocou šifrovania *MD5*, aby v databáze neboli heslá uložené v nezašifrovanej podobe.

Osobné údaje:	
Váš login:	<input type="text"/>
Heslo:	<input type="password"/>
Meno:	<input type="text"/>
Priezvisko:	<input type="text"/>
Adresa:	<input type="text"/>
Mesto:	<input type="text"/>
Kraj:	Bratislavský <input type="button" value="v"/>

Obrázok 5.1: Registrácia nového zákazníka

Po registrácii sa môže zákazník ihneď prihlásiť. Prihlásenie prebieha zadaním prihlasovacích údajov *login* a *heslo*, ktoré sú situované na ľavej strane prihlasovacej obrazovky (obrázok 5.2). Po odoslaní prihlasovacích údajov systém automaticky rozpozná podľa *loginu*, či sa jedná o zákazníka, alebo niektorého zo zamestnancov a podľa toho sa mu zobrazia aj ďalšie možnosti práce so systémom. *Heslo* je pri odosielaní automaticky zašifrované pomocou *MD5*, aby sa mohlo overiť s uloženým heslom v databáze.

► Prihlásenie

LOGIN:

HESLO:

Obrázok 5.2: Prihlásenie do systému

Po prihlásení sú zákazníkovi ponúkané tieto možnosti:

- Ponuka tovaru, kde si môže prezerat' skladom ponúkaný tovar a následne si ho objednať (obrázok 5.3)
- Prehľad objednávok, kde si zákazník môže prezerat' svoje už zadané objednávky a po prípade ich môže zmeniť, alebo stornovať (obrázok 5.4)
- Editácia osobných údajov, tu si môže zákazník prezrieť svoj profil a je mu umožnené tieto údaje aj zmeniť napr. zmena hesla, alebo zmena doručovacej adresy (obrázok 5.5)

5.3.1.1 Vytvorenie objednávky

Keď sa zákazník rozhodne prezerať sklados ponúkaný tovar, má možnosť zobrazit' tovar podľa druhu aký ho práve zaujíma. Tovar je rozdelený do týchto kategórií: *elektronika, hudba, knihy, oblečenie, šport a domáce doplnky*. Samozrejme si môže zobrazit' aj všetok tovar naraz. Objednávka sa potom zadá veľmi jednoducho. Zákazník si označí tovar o ktorý má záujem a zadá koľko kusov tovaru si chce objednať. Systém skontroluje, či je požadovaný počet kusov na sklade, ak nie, tak na to zákazníka upozorní. Keď sa objednávka prevedie, systém sa spýta, či má ešte zákazník záujem o ďalšiu objednávku, ak nie, zákazník je presmerovaný na prehľad svojich objednávok.

Označ	ID tovaru	Druh tovaru	Názov	Počet kusov na sklade	Cena tovaru
<input type="radio"/>	1	elektro	Mini notebook MSI U100 WIND biel	6	352.00 €
<input type="radio"/>	2	doplnky	Sada naradia TK-Hobby	35	12.00 €

Bližšie informácie o vybranom tovare

Objednanie tovaru:
Zadaj počet kusov:

Obrázok 5.3: Ponuka tovaru

5.3.1.2 Prehľad objednávok

Po vytvorení objednávok si zákazník môže zobrazit' prehľad svojich objednávok. Tu ich ešte môže editovať. Ak sa rozhodne zmeniť počet kusov objednaného výrobku označí si objednávku a klikne na tlačidlo *Edituj*, kde si môže zmeniť počet kusov. Ak sa rozhodne stornovať objednávku, označí si ju a klikne na tlačidlo *Zmaž*.

Označ	Druh tovaru	Názov	Značka	Množstvo	Celková cena	Dátum objednania
<input type="radio"/>	elektro	MP3 Prehrávac TEAC MP-550 8G	TEAC	1	69.00 €	2009-04-29
<input type="radio"/>	doplnky	Kancelárska stolička Perfect GTP profil	Perfect GTP	2	129.26 €	2009-05-01

Editácia objednávky:
Množstvo:
Cena:

Obrázok 5.4: Prehľad objednávok

5.3.1.3 Editácia osobných údajov

Zákazník má možnosť prezrieť si svoj profil a prípadne zmeniť svoje informácie. Medzi údaje, ktoré si zákazník môže zmeniť sú: *heslo, adresa, mesto a kraj*. Ostatné údaje sa už meniť nedajú.



Editácia osobných údajov:	
Login:	zakaznik
Heslo:
Meno:	Janko
Priezvisko:	Mrkvicka
Adresa:	Zelena 5
Mesto:	Nitra
Kraj:	Nitrianský
Počet objednávok:	1

Ulož

Obrázok 5.5: Editácia osobných údajov

Ak sa zákazník rozhodol skončiť svoju prácu v systéme a chce opustiť systém, je mu umožnené odhlásenie. Postačí kliknúť na tlačidlo odhlásiť, ktoré je situované v ľavej časti obrazovky. Systém má zabudovaný aj časovač, ktorý slúži na automatické odhlásenie používateľa pri nečinnosti dlhšej ako 15 minút.

5.3.2 Možnosti zamestnancov

Každý informačný systém musí mať aj podporu pre prácu zamestnancov. Tí tu môžu vykonávať rôzne funkcie, podľa ich zaradenia. Môj informačný systém počíta s podporou pre tri rôzne role zamestnancov: správca objednávok, skladník a vedúci skladu.

5.3.2.1 Prihlásenie do systému

Zamestnanci sa do systému prihlasujú obdobne ako zákazníci. Nepotrebuje sa však pred tým registrovať, pretože ich osobné údaje im do systému vloží správca skladu. Prihlásenie teda prebieha zadáním *loginu a hesla*. Po prihlásení si zamestnanci podľa ich práva môžu vybrať z možností, ktoré sú detailnejšie popísané nižšie.

5.3.2.2 Výpis zákaznických objednávok

Táto možnosť je sprístupnená správcovi objednávok a vedúcemu skladu. Zamestnanec si môže prezrieť zadané objednávky podľa jednotlivých zákazníkov. Zvolí si zákazníka (obrázok 5.6) a následne sa mu zobrazia jeho objednávky. Tie potom môže ešte editovať, alebo ich môže aj stornovať (obrázok 5.7).

Označ	ID Zákazníka	Login	Meno	Priezvisko	Adresa	Mesto	Kraj	Počet objednávok
	14	zak1	Elena	Bracníková	Svätoplukova 23	Pezinok	Bratislavský	2
	15	zak2	Ivan	Novosadský	Romanova 52	Piestany	Trnavský	2
	16	zak	Daniela	Orsáková	Sintavská 76	Liptovský Mikuláš	Zilinský	2
	17	zak3	Juraj	Brazinov	Jiráskova 23	Myjava	Trencianský	1
	18	zak4	Stefan	Miholko	Strecianska 91	Trnava	Trnavský	3

Vyber zákazníka

Obrázok 5.6: Výber zákazníka pre zobrazenie objednávky

Objednávky zákazníka číslo: '14'							
Označ	ID objednávky	Druh tovaru	Názov	Značka	Množstvo	Celková cena	Dátum objednania
	12	elektro	MP3 Prehrávač TEAC MP-550 8G	TEAC	1	69.00 €	2009-04-29
	13	doplňky	Kancelárska stolička Perfect GTP profil	Perfect GTP	2	129.26 €	2009-05-01

Zmena objednávky Storno objednávky

Obrázok 5.7: Objednávky vybraného zákazníka

5.3.2.3 Zobrazenie štatistík OLAP kocky

Jednotlivé tabuľky OLAP kocky si môže prezerať správca objednávok a vedúci skladu. Môže si dať zobraziť tabuľky podľa dimenzií, ktoré sú uložené v tabuľkách. Systém automaticky počíta aj sumy buď z ceny objednávok, alebo z počtu objednaných kusov. Na nasledujúcich obrázkoch je vybraná možnosť zobrazenia OLAP tabuliek pre dimenzie *Kraj*, z ktorého pochádza zákazník a *Druh objednaného tovaru*. Sumy sa budú počítať z ceny jednotlivých objednávok.

Sumy pre kraje a druhy tovarov:		
Kraj:	Druh:	Suma ceny:
Bratislavský	doplňky	129.26 €
Bratislavský	elektro	69.00 €
Trencianský	sport	583.54 €
Trnavský	doplňky	373.50 €
Trnavský	elektro	165.00 €
Trnavský	hudba	729.93 €
Trnavský	oblecenie	25.80 €
Zilinský	elektro	379.00 €
Zilinský	oblecenie	8.93 €

Obrázok 5.8: Sumy z ceny objednávok pre kraje a druhy tovarov

Sumy len pre kraje:	
Kraj:	Suma ceny:
Bratislavský	198.26 €
Trencianský	583.54 €
Trnavský	1294.23 €
Zilinský	387.93 €

Sumy len pre druhy:	
Druh:	Suma ceny:
doplňky	502.76 €
elektro	613.00 €
hudba	729.93 €
oblecenie	34.73 €
sport	583.54 €

Celková suma ceny:	
2463.96 €	

Obrázok 5.9: Sumy z ceny objednávok zvlášť pre kraj a zvlášť pre druh tovarov.

Nakoniec ešte celková suma z ceny objednávok

5.3.2.4 Objednanie tovaru na sklad

Ak sa tovar na sklade minie je povinnosťou správcu tento tovar doobjednať, alebo ho zrušiť z ponuky tovarov. Práve na to slúži táto možnosť, ktorú môže vykonávať správca objednávok a vedúci skladu.

5.3.2.5 Správa zamestnancov

Môže ju vykonávať iba vedúci skladu, žiaden iný zamestnanec k tejto ponuke nemá prístup. Vedúci skladu tu môže editovať údaje svojim zamestnancom, pridávať nových zamestnancov, alebo ich mazať z databáze. Taktiež im môže meniť platovú triedu.

5.4 Funkčnosť OLAP technológie

Pred vytvorením jednotlivých tabuliek dimenzií a faktov som si musel určiť schéma týchto dimenzií. Pre moje použitie sa najviac hodí hviezdicové schéma (vid'. kapitola 3.5.2 Schéma tabuliek dimenzií). Môj systém používa na funkčnosť OLAPu tri tabuľky dimenzií a jednu tabuľku faktov, ktoré sú uložené v databázovej štruktúre. Tabuľky dimenzií sú prepojené s tabuľkou faktov podľa identifikátorov jednotlivých dimenzií (obrázok 5.10).



Obrázok 5.10: Tabuľky dimenzií a tabuľka faktov, vytvorená podľa hviezdicovej schémy

5.4.1 Prepojenie tabuliek OLAPu s produkčnou databázou

Otázku ako prepojiť jednotlivé tabuľky OLAPu s produkčnou databázou, tak by sa tieto tabuľky boli vždy aktuálne som si kládol dlhšiu dobu. Nakoniec som sa rozhodol pre využitie triggerov, pomocou ktorých vždy pri vkladaní, aktualizovaní, alebo mazaní dát z produkčnej databázy sa automaticky aktualizujú dáta aj v tabuľkách OLAPu.

5.4.1.1 Typy triggerov

Triggerov som musel vytvoriť viacero a to podľa ich funkcie a podľa toho, ktorú tabuľku v produkčnej databáze sledovali. Jeden zabezpečuje aktuálnosť OLAP tabuliek pri vkladaní dát, ďalší pri aktualizovaní dát a ďalší pri mazaní dát. Ako príklad som uviedol trigger, ktorý vloží dáta do OLAP tabuliek pri zadaní novej objednávky.

```
CREATE TRIGGER `xnagyd00`.`ins_objednavka` AFTER INSERT ON
`xnagyd00`.`objednavka`
FOR EACH ROW BEGIN
if (SELECT datum FROM olap_cas WHERE datum = new.date_obj) is null then
INSERT INTO `xnagyd00`.`olap_cas`
VALUES (null, new.date_obj, day( new.date_obj ), month( new.date_obj ),
extract(quarter from new.date_obj), year( new.date_obj ));
end if;
INSERT INTO olap_fakty VALUES((SELECT id_dat from olap_cas WHERE
datum=new.date_obj), new.id_tovar, new.id_zakaznik, new.id_objednavka,
new.celk_poc_tov, new.celk_cena_obj);
END
```

Pomocou tohto triggeru sa do časovej dimenznej tabuľky vloží *dátum, deň, mesiac, kvartál a rok* iba vtedy, keď takýto dátum ešte v tabuľke nie je uvedený. Ďalej sa do tabuľky faktov vloží identifikátor *časovej a produktovej* dimenzie, *identifikátor, počet kusov* a *cena* objednávky.

5.4.2 Grafická reprezentácia OLAPu

Keďže OLAP sa využíva ako štatistický prostriedok, je potrebné aby údaje z neho mali nejakú zrozumiteľnú podobu. Modul, ktorý som navrhol a implementoval získava údaje z tabuliek OLAPu zobrazuje ich v prehľadných tabuľkách podľa zadaných kritérií používateľa informačného systému. Príklad tohto zobrazenia je uvedený v kapitole 5.3.2.3 *Zobrazenie štatistík OLAP kocky*.

Pred tým ako sú dáta z OLAP tabuliek zobrazené, treba ich najskôr pospájať (dáta sú uložené vo viacerých tabuľkách dimenzií) a pretriediť, aby vyhovovali kritériám zadaných používateľom. Ja som k tomuto účelu využil zložitejšie MySQL dotazy k spojovaniu tabuliek, zoskupovaniu a zoradovaniu záznamov, ktoré sa v nich nachádzajú. Jeden takýto dotaz ukážem na príklade:

```
SELECT olap_produkt.druh, olap_zakaznik.obvod, SUM(olap_fakty.cena) AS cena
FROM olap_fakty
INNER JOIN olap_produkt ON olap_produkt.id_produkt = olap_fakty.id_produkt
INNER JOIN olap_zakaznik ON olap_zakaznik.id_zakaznik = olap_fakty.id_zakaznik
GROUP BY olap_zakaznik.obvod, olap_produkt.druh
```

ORDER BY olap_zakaznik.obvod, olap_produkt.druh

V tomto príklade načítavam dáta z dvoch tabuliek dimenzií (*druh tovaru a kraj z ktorého pochádza zadávateľ objednávky*) a z tabuľky faktov (*cena objednávky*). Dimenzie teda podľa identifikátorov nie je problém spojiť s tabuľkou faktov. Potom sa už len načítané dáta zoskupia, zoradia a vypočítajú sumy. V takomto tvare sa už môžu dáta graficky zobraziť (*vid'. kapitola 5.3.2.3 Zobrazenie štatistík OLAP kocky*) a použiť aj na štatistické účely.

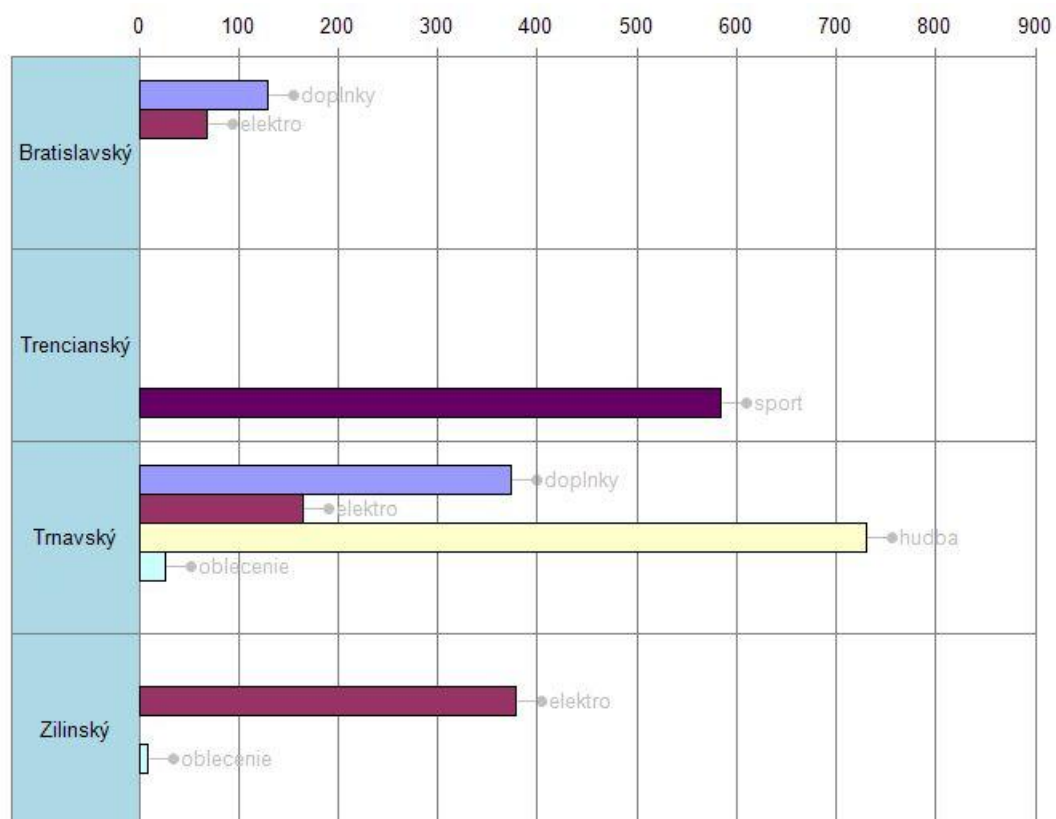
Pre profesionálne zobrazovanie štatistík pomocou OLAPu sa však v dnešnej dobe používajú rôzne analytické manažéry, ktoré sa dokážu pripojiť priamo na databázu s uloženou OLAP kockou a zobraziť potrebné informácie. Pre názornosť funkčnosti OLAPu som si vybral jeden z voľne šíriteľných manažérov, s podporou pripojenia k MySQL databáze. Jeho názov je OlapCube. Pomocou neho som si dal zobraziť dáta (štatistiky) s rovnakými kritériami aké som určil v kapitole 5.3.2.3 *Zobrazenie štatistík OLAP kocky*. Na obrázku 5.11 je zobrazená tabuľka so štatistickými údajmi pre kraj, druh tovaru a sumami z cien objednávok.

	doplňky	elektro	hudba	oblecenie	sport	Druh tovaru
Bratislavský	129,26	69,00				198,26
Trencianský					583,54	583,54
Trnavský	373,50	165,00	729,93	25,80		1 294,23
Zilinský		379,00		8,93		387,93
Kraj	502,76	613,00	729,93	34,73	583,54	2 463,96

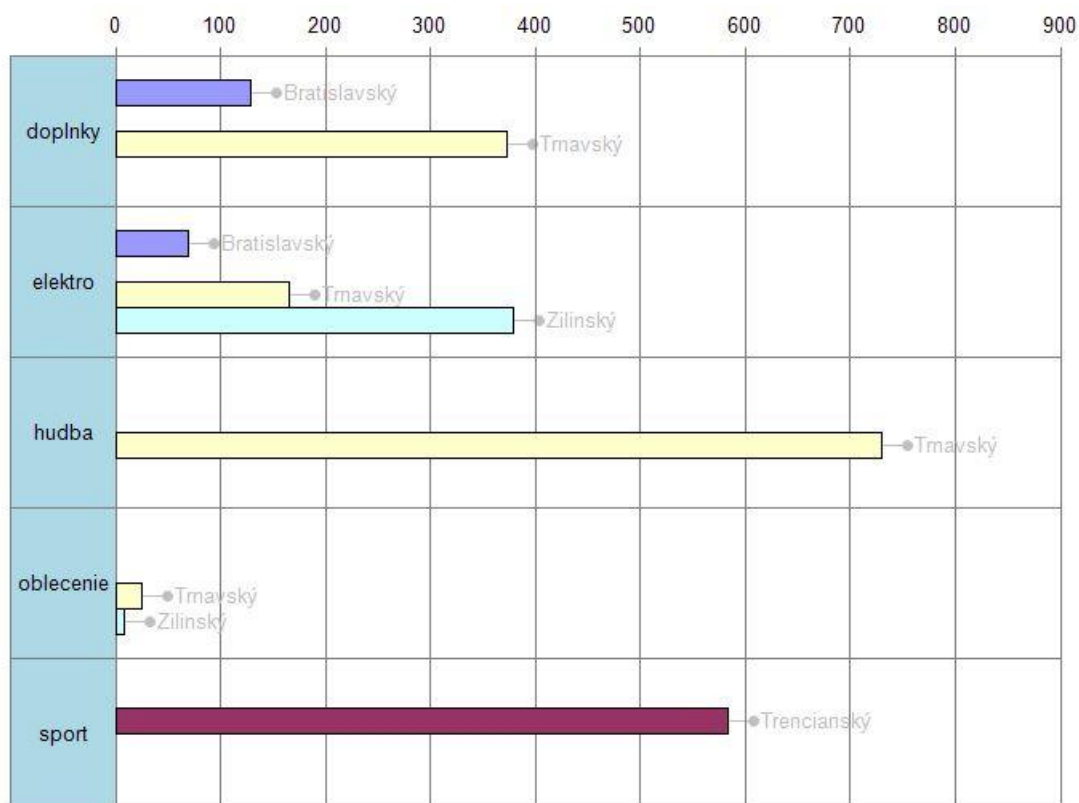
Obrázok 5.11: Tabuľka štatistík zobrazená pomocou OlapCube

Samozrejme tu nechýba ani podpora grafov. Na obrázkoch 5.12 a 5.13 uvediem grafy s rovnakými kritériami ak pri predchádzajúcej tabuľke.

Na efektívnu správu skladu je dobré mať čo najviac informácií, získaných zo štatistík, aby sa vedenie mohlo napríklad dopredu predzásobiť určitým typom tovaru, ktorý si zákazníci v určitom období najviac objednávajú. Pomocou technológie OLAP je tak správa štatistík rýchlejšia prehľadnejšia a jednoduchšia.



Obrázok 5.12: Graf štatistík vygenerovaný pomocou OlapCube



Obrázok 5.13: Graf štatistík vygenerovaný pomocou OlapCube

6 Záver

Navrhovaný a implementovaný systém, ktorý som vytvoril ako bakalársku prácu splňuje požiadavky uvedené v zadaní. Dôraz som kládol na jednoduchosť ovládania systému a príjemné užívateľské rozhranie, v ktorom sú implementované funkcie pre registráciu nových zákazníkov, prihlásenie do systému, zmena osobných údajov, prehľad ponuky tovarov, zadanie objednávky a iné.

Systém dokáže pracovať s rôznymi typmi účtov, môžu sa tam teda prihlásiť aj zamestnanci, konkrétnejšie správcovia objednávok, skladníci a vedúci skladu. Taktiež majú aj rôzne možnosti v systéme a to cez prehľad zákazníkov, editáciu ich objednávok, objednanie nového tovaru na sklad, príjem a výdaj objednávok a zobrazenie štatistík z OLAP kocky. Vedúci skladu ešte môžu prijímať, alebo vyhadzovať zamestnancov a meniť platovú triedu.

Počas vývoja som testoval systém na rôznych webových prehliadačoch, konkrétnejšie: Mozilla Firefox, Opera, Google Chrome a Internet Explorer 7, v ktorom sa ale často objavovali menšie nezrovnalosti v zobrazovaní, ktoré som ale vyriešil.

Systém pracuje aj s technológiou OLAP, pri ktorej implementovaní do systému, som uvažoval o rôznych modeloch, ktoré by zabezpečovali previazanosť s produkčnou databázou. Nakoniec som sa rozhodol pre trigger, ktoré automaticky plnia, aktualizujú, alebo mažu dáta z OLAP tabuliek pri zmenách v produkčnej databáze.

V súčasnej dobe je systém pripravený na používanie. Chyby, ktoré neboli odhalené sa môžu časom prejaviť. Taktiež by bolo potrebné systém prispôbiť na mieru konkrétnej firme, ktorá by o neho mala záujem.

Ďalší vývoj môjho systému by sa mohol týkať v rozšírení možností pre zákazníka napríklad sledovanie zmien ceny produktov. Taktiež by sa systém mohol rozšíriť v sledovaní a zobrazovaní väčšieho množstva štatistík, teda rozšíriť OLAP kocku o viac dimenzií.

Bakalárska práca mi vylepšila znalosti o informačných systémoch, v práci s databázou a vďaka nej som sa oboznámil aj s novými technológiami, akou je OLAP a jeho praktické využitie pri sledovaní rôznych štatistík. Pomohla mi, aby som bol pripravený vstúpiť do náročnejšieho, ale nesmierne zaujímavého procesu vývoja softwarového produktu.

Literatúra

- [1] NAGY, Peter. Úvod do informačných systémov. *Informačné systémy*. 2003, s. 8.
- [2] *Wikipédia, slobodná encyklopédia* [online]. 2009 [cit. 2009-05-02]. Dostupný z WWW: <http://sk.wikipedia.org/wiki/Hypertext_markup_language>.
- [3] *Wikipédia, slobodná encyklopédia* [online]. 2009 [cit. 2009-05-03]. Dostupný z WWW: <[http://sk.wikipedia.org/wiki/PHP_\(programovac%C3%AD_jazyk\)](http://sk.wikipedia.org/wiki/PHP_(programovac%C3%AD_jazyk))>.
- [4] ZAJÍC, Petr. *Linuxsoft* [online]. 2005 [cit. 2009-05-06]. Dostupný z WWW: <http://www.linuxsoft.cz/article.php?id_article=731>.
- [5] LACKO, Luboslav. *Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle*. 1. vyd. .Brno : Computer Press, 2003. 486 s. ISBN 8072269690.

Zoznam príloh

Príloha 1. CD